

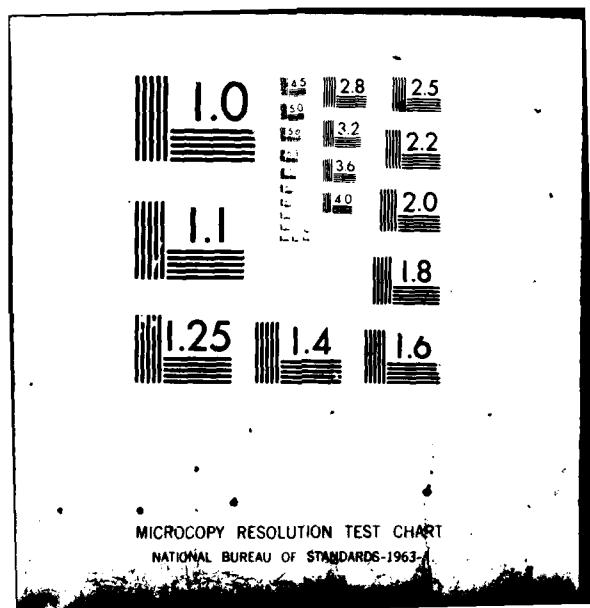
AD-A085 458

GEORGIA INST OF TECH ATLANTA SCHOOL OF INDUSTRIAL AN--ETC F/6 12/1
AN ALGORITHM FOR LINEARLY CONSTRAINED NONLINEAR PROGRAMMING PRO--ETC(U)
1980 M S BAZARA, J J GOODE F49620-79-C-0120
AFOSR-TR-80-0449 NL

UNCLASSIFIED

100-1
40-
200-100

END
DATE
FILED
7-80
DTC



ADA085458

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR/TR- 80-0449	2. GOVT ACCESSION NO. AD-A082458	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN ALGORITHM FOR LINEARLY CONSTRAINED NONLINEAR PROGRAMMING PROBLEMS	5. TYPE OF REPORT & PERIOD COVERED Interim ref to	
6. AUTHOR(S) Mokhtar S. Bazaraa Jamie J. Goode	7. PERFORMING ORGANIZATION NAME AND ADDRESS Georgia Institute of Technology School of Industrial & Systems Engineering Atlanta, GA 30332	
8. CONTRACT OR GRANT NUMBER(S) F49620-79-C-0128 Rev		9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 17A6 61103E 2304/A6
10. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, Washington, DC 20332	11. REPORT DATE 11/1980	
12. NUMBER OF PAGES 21	13. SECURITY CLASS. (of this report) UNCLASSIFIED	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		
15. DECLASSIFICATION/DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) In this paper an algorithm for solving a linearly constrained nonline programming problem is developed. Given a feasible point, a correction vector is computed by solving a least distance programming problem over a polyhedral cone defined in terms of the gradients of the *almost* binding constraints. Mukai's approximate scheme for computing step sizes is generalized to handle the constraints. This scheme provides an estimate for the step size based on a quadratic approximation of the function. This estimate is used in conjunction with		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Abstract cont'

Armijo line search to calculate a new point. It is shown that each accumulation point is a Kuhn-Tucker point to a slight perturbation of the original problem. Furthermore, under suitable second order optimality conditions, it is shown that eventually only one trial is needed to compute the step size.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

9

DTIC
SELECTED
S JUN 12 1980 D
C

AN ALGORITHM FOR LINEARLY CONSTRAINED
NONLINEAR PROGRAMMING PROBLEMS

by

Mokhtar S. Bazaraa

and

Jamie J. Goode

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TECHNICAL APPROVAL
This technical report has been reviewed and is
approved for public release and sale under AFRL-AR-100-12 (7b).
Distribution is unlimited.
A. D. BROWN
Technical Information Officer

This document has been approved
for public release and sale; its
distribution is unlimited.

AN ALGORITHM FOR LINEARLY CONSTRAINED
NONLINEAR PROGRAMMING PROBLEMS

Mokhtar S. Bazaraa[†] and Jamie J. Goode^{††}

In this paper an algorithm for solving a linearly constrained nonlinear programming problem is developed. Given a feasible point, a correction vector is computed by solving a least distance programming problem over a polyhedral cone defined in terms of the gradients of the "almost" binding constraints. Mukai's approximate scheme for computing step sizes is generalized to handle the constraints. This scheme provides an estimate for the step size based on a quadratic approximation of the function. This estimate is used in conjunction with Armijo line search to calculate a new point. It is shown that each accumulation point is a Kuhn-Tucker point to a slight perturbation of the original problem. Furthermore, under suitable second order optimality conditions, it is shown that eventually only one trial is needed to compute the step size.

[†] School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia. This author's work is supported under USAFOSR contract number F49620-79-C-0120.

^{††} School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia.

Accession No.	7715 00001	Classification	Approved for public release	Avail and/or special
Author	DOC 103	By	Justification	Ref
Editor	Unnumbered	Disposition	Comments	Plat

1. Introduction

This paper addresses the following linearly constrained nonlinear programming problem:

$$\begin{aligned} P: \quad & \text{minimize} && f(x) \\ & \text{subject to} && Ax \leq b \end{aligned}$$

where f is a twice continuously differentiable function on \mathbb{R}^n , and A is an $l \times n$ matrix whose j th row is denoted by a_j^t , and where a superscript t denotes the transpose operation.

There are several approaches for solving this problem. The first one relies on partitioning the variables into basic, nonbasic, and superbasic variables. The values of the superbasic and basic variables are modified while the nonbasic variables are fixed at their current values. Examples of methods in this class are the convex simplex method of Zangwill [18], the reduced gradient method of Wolfe [17], the method of Murtagh and Saunders [12], and the variable reduction method of McCormick [8].

Another class of methods is the extension of quasi-Newton algorithms from unconstrained to constrained optimization. Here, at any iteration, a set of active restrictions is identified, and then a modified Newton procedure is used to minimize the objective function on the manifold defined by these active constraints. See for example Goldfarb [6], and Gill and Murray [5].

Other approaches for solving problems with linear constraints are the gradient projection method and the method of feasible directions. The former computes a direction by projecting the negative gradient on the space orthogonal to the gradients of a subset of the binding constraints while the latter method determines a search direction by solving a linear programming problem.

For a review of these methods the reader may refer to Rosen [14], Zoutendijk [19], Frank and Wolfe [4], and Topkis and Veinott [15].

In this paper, an algorithm for solving problem P is proposed. At each iteration a correction vector is computed by finding the minimum distance from a given point to a polyhedral cone defined in terms of the gradients of the "almost" binding constraints. An approximate line search procedure which extends those of Armijo [1] and Mukai [10, 11] for unconstrained optimization is developed for determining the step size. First, an estimate of the step size based on a quadratic approximation to the objective function is computed, and then adjusted if necessary.

In Section 2, we outline the algorithm. In Section 3, we show that accumulation points of the algorithm are Kuhn-Tucker points to a slight perturbation of the original problem. Finally, in Section 4, assuming that the algorithm converges, and under suitable second order sufficiency optimality conditions, we show that the step size estimates which are based on the quadratic approximation are acceptable so that only one functional evaluation is eventually needed for performing the line search.

2. Statement of the Algorithm

Consider the following algorithm for solving Problem P.

Step 0

Choose values for the parameters c , z , δ , and ϵ . Select a point x_0 such that $Ax_0 \leq b$ and let $\delta_0 = \delta$. Let $i = 0$ and go to Step 1.

Step 1

Let w_i be the optimal solution to Problem $D(x_i)$ given below:

$$D(x_i): \text{minimize } \nabla f(x_i)^t w + \frac{1}{2} z w^t w \\ \text{subject to } a_j^t w \leq 0 \quad \text{for } j \in I(x_i)$$

where

$$I(x_i) = \{j: a_j^t x_i > b_j - c\} \quad (2.1)$$

If $w_i = 0$, stop. Else, go to Step 2.

Step 2

Let

$$I^+(w_i) = \{j: a_j^t w_i > 0\} \quad (2.2)$$

and let

$$\beta_i = \min \{1, \frac{b_j - a_j^t x_i}{a_j^t w_i} \text{ for } j \in I^+(w_i)\} \quad (2.3)$$

Let

$$d_i = \beta_i w_i \quad (2.4)$$

and go to Step 3.

Step 3

If

$$f(x_i + \varepsilon d_i) + f(x_i - \varepsilon d_i) - 2f(x_i) \geq \varepsilon^2 \delta_i \|d_i\|^2 \quad (2.5)$$

let

$$\lambda_i = \frac{-\varepsilon^2 \nabla f(x_i)^t d_i}{f(x_i + \varepsilon d_i) + f(x_i - \varepsilon d_i) - 2f(x_i)} \quad (2.6)$$

and let $\delta_{i+1} = \delta_i$, and go to Step 4. Otherwise, let $\lambda_i = 1$, $\delta_{i+1} = \frac{1}{2} \delta_i$, and go to Step 4.

Step 4

Let

$$\alpha_i = \min \{1, \lambda_i\} \quad (2.7)$$

and compute the smallest nonnegative integer k satisfying

$$f(x_i + (\frac{1}{2})^k \alpha_i d_i) - f(x_i) \leq \frac{1}{3} (\frac{1}{2})^k \alpha_i \nabla f(x_i)^T d_i \quad (2.8)$$

Let $k_i = k$, $x_{i+1} = x_i + \alpha_i (\frac{1}{2})^k d_i$, $i = i + 1$, and go to Step 1.

The following remarks are helpful in interpreting the above algorithm.

1. A direction w_i is determined by solving Problem $D(x_i)$. This problem finds the point in the convex polyhedral cone $\{w: a_j^T w \leq 0 \text{ for } j \in I(x_i)\}$ which is closest to the vector $-\frac{1}{z} \nabla f(x_i)$. Methods of least distance programming, as in the works of Bazaraa and Goode [2], and Wolfe [16] can be used for solving this problem. Special methods that take advantage of the structure of the cone constraints may prove quite useful in this regard.

2. The restrictions enforced in Problem $D(x_i)$ are the c-binding constraints at x_i , that is, those satisfying $b_j - c < a_j^T x_i \leq b_j$. If $w_i = 0$, then the algorithm is terminated with x_i . In this case, from the Kuhn-Tucker conditions for Problem $D(x_i)$, there exist u_j for $j \in I(x_i)$ such that:

$$Vf(x_i) + \sum_{j \in I(x_i)} u_j a_j = 0$$

$$u_j \geq 0 \quad \text{for } j \in I(x_i)$$

These conditions imply that x_i is a Kuhn-Tucker point for the following problem:

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && a_j^T x \leq a_j^T x_i \quad \text{for } j \in I(x_i) \\ & && a_j^T x \leq b_j \quad \text{for } j \notin I(x_i) \end{aligned}$$

Noting that $b_j - c < a_j^T x_i \leq b_j$ for $j \in I(x_i)$, if c is sufficiently small, it is clear that the algorithm is terminated if x_i is a Kuhn-Tucker solution to a slightly perturbed version of Problem P. The following definition will thus be useful.

Definition 2.1

Let x^* be a feasible point to Problem P. If the optimal solution to Problem $D(x^*)$ is equal to zero, then x^* is called a c-KT solution to Problem P.

3. If $x_i + w_i$ is feasible to Problem P, then the search vector d_i is taken as w_i . Otherwise, d_i is taken to be the vector of maximum length along w_i which maintains feasibility of $x_i + d_i$.

4. Steps 3 and 4 of the algorithm compute the step size taken along the vector d_i in order to form x_{i+1} . As proposed by Mukai [10, 11], first an estimate of the step size λ_i is calculated. When appropriate, λ_i is computed by utilizing a quadratic approximation of the function f at x_i , otherwise λ_i is taken equal to 1. In order to ensure feasibility to Problem P, the first trial step size α_i used in conjunction with Armijo line search [1], is the minimum of λ_i and 1. As will be shown in Section 4, under suitable assumptions, for large i , test (2.5) passes, $k_i = 0$, and $\alpha_i = \lambda_i < 1$. This confirms efficiency of the line search scheme where eventually only one trial is needed to compute the step size.

3. Accumulation Points of the Algorithm

Theorem 3.1 shows that each accumulation point of the proposed algorithm is a c-KT point. In order to prove this theorem, lemmas 3.1 and 3.2 are needed. These two lemmas extend similar results of Mukai [10] for unconstrained problems.

In order to facilitate the development in this section, the following notation is used. Let $w(x)$ be the optimal solution to Problem $D(x)$ and let $\beta(x)$ be as given in (2.3) with x_i replaced with x . Finally, let $d(x) = \beta(x)w(x)$.

Lemma 3.1

Suppose that x^* is not a c-KT point for Problem P. Then, there exist scalars μ and $s > 0$ so that $\mu \leq \alpha(x) \leq 1$ for each x with $\|x-x^*\| < s$.

Proof

There exists $s_1 > 0$ so that $I(x) = I(x^*)$ for all $\|x-x^*\| < s_1$. Thus, the feasible region for Problem $D(x)$ is equal to that of Problem $D(x^*)$ for all x satisfying $\|x-x^*\| < s_1$. By continuous differentiability of f , it then follows that $w(\cdot)$ is continuous in x at x^* , see for example Daniel [3]. Particularly,

there exists a number $s_2 > 0$ such that $I^+(w(x)) = I^+(w(x^*))$ if $\|x-x^*\| < s_2$.

This together with the continuity of $w(\cdot)$ and the formula for computing $\beta(\cdot)$ imply that $\beta(\cdot)$ is continuous in x at x^* . Hence, $d(\cdot)$ is also continuous.

Since x^* is not a c-KT point, then $w(x^*) \neq 0$. Furthermore, $b_j - a_j^t x^* \geq c$ if $a_j^t w > 0$ which implies that $\beta(x^*) > 0$. Therefore $d(x^*) \neq 0$. By continuity of $\beta(\cdot)$ and $d(\cdot)$ at x^* there exist scalars q and $s > 0$ so that

$$\beta(x) \|d(x)\|^2 \geq \frac{1}{2} \beta(x^*) \|d(x^*)\|^2 \quad \text{if } \|x-x^*\| < s \quad (3.1)$$

$$f(x+\epsilon d(x)) + f(x-\epsilon d(x)) - 2f(x) < q \quad \text{if } \|x-x^*\| < s \quad (3.2)$$

Now, let x be such that $\|x-x^*\| < s$. Since $v(x)$ solves Problem D(x), then $\nabla f(x)^t w(x) \leq -\frac{1}{2} \epsilon \|w(x)\|^2$. This, in turn, implies that $-\nabla f(x)^t d(x) \geq \frac{1}{2} \epsilon \beta(x) \|d(x)\|^2$ and from (3.1) we get:

$$-\nabla f(x)^t d(x) \geq \frac{1}{4} \epsilon \beta(x^*) \|d(x^*)\|^2 = v > 0 \quad (3.3)$$

If test (2.5) passes, then from (3.2) and (3.3) the following lower bound on λ_i is at hand:

$$\lambda_i = \frac{-\epsilon^2 \nabla f(x)^t d(x)}{f(x+\epsilon d(x)) + f(x-\epsilon d(x)) - 2f(x)} \geq \frac{\epsilon^2 v}{q}$$

If test (2.5) fails, then $\lambda_i = 1$ and hence $\lambda_i \geq \min\{1, \frac{\epsilon^2 v}{q}\} = \mu$. Since $\alpha_i = \min\{1, \lambda_i\}$, the desired result follows.

Lemma 3.2

If x^* is not a c-KT point for Problem P, then there exist a number $s > 0$ and an integer m so that $k(x) \leq m$ if $\|x-x^*\| < s$, where $k(x)$ is the Armijo integer

given by (2.8) with x_i and α_i replaced with x and $\alpha(x)$ respectively.

Proof

As in the proof of Lemma 3.1 and by continuous differentiability of f , there exist scalars s , h , and $y > 0$ so that for $\|x - x^*\| < s$ the following hold:

$$\nabla f(x)^T d(x) \leq -y \quad (3.4)$$

$$|\nabla f(x+gd(x))^T d(x) - \nabla f(x)^T d(x)| \leq \frac{2}{3} y \quad \text{for each } g \in [0, h] \quad (3.5)$$

Now let m be the smallest nonnegative integer so that $(\frac{1}{2})^m \leq h$ and let x be such that $\|x - x^*\| < s$. Then there exists $\theta \in [0, 1]$ such that:

$$\begin{aligned} f(x + (\frac{1}{2})^m \alpha(x) d(x)) - f(x) &= \frac{1}{3} (\frac{1}{2})^m \alpha(x) \nabla f(x)^T d(x) \\ &= (\frac{1}{2})^m \alpha(x) \nabla f(x + \theta (\frac{1}{2})^m \alpha(x) d(x))^T d(x) - \frac{1}{3} (\frac{1}{2})^m \alpha(x) \nabla f(x)^T d(x) \\ &= (\frac{1}{2})^m \alpha(x) \left[\{ \nabla f(x + \theta (\frac{1}{2})^m \alpha(x) d(x))^T d(x) - \nabla f(x)^T d(x) \} + \frac{2}{3} \nabla f(x)^T d(x) \right] \quad (3.6) \end{aligned}$$

Since $\theta (\frac{1}{2})^m \alpha(x) \leq h$, (3.4) and (3.5) imply that the right hand side of (3.6) is ≤ 0 which in turn shows that $k(x) \leq m$, and the proof is complete.

Theorem 3.1

Either the algorithm terminates with a c-KT point for Problem P or else generates an infinite sequence $\{x_i\}$ of which any accumulation point is a c-KT point for Problem P.

Proof

Clearly the algorithm stops at x_i only if x_i is a c-KT point. Now, suppose that the algorithm generates the infinite sequence $\{x_i\}$. Suppose that x^* is an accumulation point so that $x_i \xrightarrow{K} x^*$ for some infinite set K of positive integers. Since $f(x_i)$ is decreasing monotonically and since $f(x_i) \xrightarrow{K} f(x^*)$ then $f(x_i) \rightarrow f(x^*)$. Suppose by contradiction to the desired conclusion that x^* is not a c-KT point. From Lemmas 3.1 and 3.2, there exist positive numbers μ and y and an integer m so that $\alpha_i \geq \mu$, $\nabla f(x_i)^T d_i \leq -y$, and $k_i \leq m$ for large i in K . Therefore,

$$f(x_{i+1}) - f(x_i) \leq \frac{1}{3} \left(\frac{1}{2}\right)^{k_i} \alpha_i \nabla f(x_i)^T d_i \leq -\frac{1}{3} \mu y \left(\frac{1}{2}\right)^m$$

for large i in K . This implies that $f(x_i) \rightarrow -\infty$, contradicting the fact that $f(x_i) \rightarrow f(x^*)$. This completes the proof.

4. Eventual Acceptance of the Step Size Estimate

In the previous section, we showed that an accumulation point x^* of the sequence $\{x_i\}$ generated by the algorithm is a KT point to the perturbed problem P' given below:

$$\begin{aligned} P': \text{minimize} \quad & f(x) \\ \text{subject to} \quad & a_j^T x \leq a_j^T x^* \quad \text{for } j \in I(x^*) \\ & a_j^T x \leq b_j \quad \text{for } j \notin I(x^*) \end{aligned}$$

Here, we assume that the whole sequence $\{x_i\}$ converges to a point x^* which satisfies suitable second order sufficiency conditions. Under this assumption, we show that test (2.5) is eventually passed. Furthermore, we show

that $\lambda_i < 1$ and that $k_i = 0$ for i large enough.

The second order condition is given in Definition 4.1. It is well-known that x^* satisfying this condition is a strong local minimum for problem P' . That is, there exists a number $\gamma > 0$ so that $f(x^*) < f(x)$ if x is feasible to problem P' and $\|x - x^*\| < \gamma$, see for example McCormick [9] and Han and Mangasarian [7].

Definition 4.1

Let x^* be such that $Ax^* \leq b$ and let $I(x^*) = \{j: a_j^t x^* > b_j - \epsilon\}$. x^* is said to satisfy the second order sufficiency optimality conditions for problem P' if there exist scalars $u_j \geq 0$ for $j \in I(x^*)$ and $\gamma > 0$ so that:

$$\nabla f(x^*) + \sum_{j \in I(x^*)} u_j a_j = 0$$

$$f(x^*)^t d \leq 0, a_j^t d \leq 0 \text{ for } j \in I(x^*), \|d\| = 1 \Rightarrow d^t H(x^*) d > \gamma \quad (4.1)$$

Theorem 4.1 shows that test (2.5) will eventually be passed so that λ_i is given by (2.6). The following two intermediate results are needed to prove this theorem.

Lemma 4.1

If $Cd \leq 0$ and $\|d\| = 1$ imply that $d^t Hd \geq \gamma > 0$ then there is a number $\theta > 0$ so that $Cd \leq \theta 1$ and $\|d\| = 1$ imply that $d^t Hd \geq \gamma/2$.

Proof

Suppose by contradiction that for each integer k there is a vector d_k such that

$$\|d_k\| = 1, Cd_k \leq \frac{1}{k} 1, \text{ and } d_k^T Hd_k < \gamma/2 \quad (4.2)$$

Since the sequence $\{d_k\}$ is bounded, it has an accumulation point d . From (4.2), $\|d\| = 1$, $Cd \leq 0$, and $d^T Hd \leq \gamma/2$ which contradicts the assumption of the lemma.

Lemma 4.2

If either $\{x_i\}$ converges or $\{x: Ax \leq b, f(x) \leq f(x_0)\}$ is bounded, then $d_i \rightarrow 0$.

Proof

Since $0 < \beta_i \leq 1$ and $d_i = \beta_i w_i$, it suffices to prove that $w_i \rightarrow 0$. Suppose there exist an infinite set of positive integers K and a number $\epsilon > 0$ so that

$$\|w_i\| \geq \epsilon \quad \text{for } i \in K \quad (4.3)$$

Clearly, under either of the assumptions of the lemma, there exist an infinite set $K' \subset K$ and a point x^* so that $x_i \xrightarrow{K'} x^*$. By Theorem 3.1 x^* is a c-KT point for Problem P. Thus, $w^* = 0$ is the unique optimal solution to Problem $D(x^*)$. But for large $i \in K'$, $I(x^*) = I(x_i)$, and by continuity of the solutions to $D(\cdot)$ we must have $\|w_i\| < \epsilon/2$ for large i in K' . This contradicts (4.3) and the proof is complete.

Throughout the remainder of this section, the following notation will be used for any scalar γ :

$$H_1^Y = 2 \int_0^1 (1-y) H(x_1 + y\gamma d_1) dy \quad (4.3)$$

We can integrate by parts to obtain

$$f(x_1 + \gamma d_1) - f(x_1) = \gamma \nabla f(x_1)^T d_1 + \frac{1}{2} \gamma^2 d_1^T H_1^Y d_1 \quad (4.4)$$

For further details, the reader may refer to Polak [13, p. 293].

Theorem 4.1

Let $\{x_i\}$ be a sequence generated by the algorithm. Suppose that $x_i \rightarrow x^*$ and x^* satisfies the second order optimality conditions for problem P' . Then there exists an integer m so that test (2.5) passes for all $i \geq m$.

Proof

From (4.3) and (4.4) we get:

$$f(x_1 + \epsilon d_1) - f(x_1) = \epsilon \nabla f(x_1)^T d_1 + \frac{1}{2} \epsilon^2 d_1^T H_1^{\epsilon} d_1$$

$$f(x_1 - \epsilon d_1) - f(x_1) = -\epsilon \nabla f(x_1)^T d_1 + \frac{1}{2} \epsilon^2 d_1^T H_1^{-\epsilon} d_1$$

Adding we obtain:

$$f(x_1 + \epsilon d_1) + f(x_1 - \epsilon d_1) - 2f(x_1) = \frac{1}{2} \epsilon^2 d_1^T (H_1^{\epsilon} + H_1^{-\epsilon}) d_1 \quad (4.5)$$

Now for $j \in I(x^*)$, $a_j^T x^* > b_j - c$. Since $x_i \rightarrow x^*$ then for i large enough, $a_j^T x_i > b_j - c$ so that $j \in I(x_i)$. By step 1 of the algorithm $a_j^T w_i \leq 0$ and so $a_j^T \frac{d_i}{\|d_i\|} \leq 0$ for i large enough and $j \in I(x^*)$. Likewise, from step 1 of the

algorithm $\nabla f(x_i)^T w_i \leq 0$ and hence $\nabla f(x_i)^T \frac{d_i}{\|d_i\|} \leq 0$. Since $x_i \rightarrow x^*$, then for any number $\theta > 0$, $\nabla f(x^*)^T \frac{d_i}{\|d_i\|} \leq \theta$ for i large enough. Thus, Lemma 4.1 and the second order conditions imply that

$$d_i^T H(x^*) d_i \geq \frac{\gamma}{2} \|d_i\|^2 \quad \text{for large } i \quad (4.6)$$

Now note that

$$\begin{aligned} \|H_i^\epsilon - H(x^*)\| &= \left\| 2 \int_0^1 (1-y) \left[H(x_i + y\epsilon d_i) - H(x^*) \right] dy \right\| \\ &\leq 2 \int_0^1 (1-y) \|H(x_i + y\epsilon d_i) - H(x^*)\| dy \end{aligned} \quad (4.7)$$

Since $x_i \rightarrow x^*$, then by Lemma 4.2, $d_i \rightarrow 0$. Particularly, for i large enough, $\|H(x_i + y\epsilon d_i) - H(x^*)\| < \frac{\gamma}{4}$ for all $y \in [0,1]$. From (4.7), $\|H_i^\epsilon - H(x^*)\| < \frac{\gamma}{4}$. This together with (4.6) yields:

$$\begin{aligned} d_i^T H_i^\epsilon d_i &= d_i^T H(x^*) d_i + d_i^T (H_i^\epsilon - H(x^*)) d_i \\ &\geq \frac{\gamma}{2} \|d_i\|^2 - \|d_i\|^2 \|H_i^\epsilon - H(x^*)\| \\ &\geq \frac{\gamma}{4} \|d_i\|^2 \quad \text{for large } i \end{aligned} \quad (4.8)$$

Similarly,

$$d_i^T H_i^{-\epsilon} d_i \geq \frac{\gamma}{4} \|d_i\|^2 \quad \text{for large } i \quad (4.9)$$

From (4.5), (4.8), and (4.9) it immediately follows that

$$f(x_i + \epsilon d_i) + f(x_i - \epsilon d_i) - 2f(x_i) \geq \epsilon^2 \frac{\gamma}{4} \|d_i\|^2 \quad \text{for large } i \quad (4.10)$$

From (4.10), if test (2.5) fails for a large i , we must have:

$$\epsilon^2 \delta_i \|d_i\|^2 > f(x_i + \epsilon d_i) + f(x_i - \epsilon d_i) - 2f(x_i) \geq \epsilon^2 \frac{\gamma}{4} \|d_i\|^2$$

that is, $\delta_i > \frac{\gamma}{4}$. If the conclusion of the lemma does not hold, then test (2.5) fails infinitely often and then $\delta_i \rightarrow 0$. This contradicts $\delta_i > \frac{\gamma}{4}$ for large i , and the proof is complete.

Theorem 4.2

Let $\{x_i\}$ be a sequence generated by the algorithm. Suppose that $x_i \rightarrow x^*$ and that x^* satisfies the second order optimality conditions for Problem P'. Then there exists an integer m so that $f(x_i + \alpha_i d_i) - f(x_i) \leq \frac{1}{3} \alpha_i \nabla f(x_i)^T d_i$ for all $i \geq m$, that is, $k_i = 0$ for all $i \geq m$.

Proof

By Theorem (4.1), test (2.5) passes for large i so that λ_i is given by

$$\lambda_i = \frac{-\epsilon^2 \nabla f(x_i)^T d_i}{f(x_i + \epsilon d_i) + f(x_i - \epsilon d_i) - 2f(x_i)} = \frac{-\nabla f(x_i)^T d_i}{\frac{1}{2} d_i^T (H_i^\epsilon + H_i^{-\epsilon}) d_i} \quad (4.11)$$

If $\lambda_i \leq 1$ so that $\alpha_i = \lambda_i$, then from (4.4) and (4.11) we get:

$$\begin{aligned} f(x_i + \alpha_i d_i) - f(x_i) - \frac{1}{3} \alpha_i \nabla f(x_i)^T d_i &= \frac{1}{2} \lambda_i^2 d_i^T H_i^\epsilon d_i + \frac{2}{3} \lambda_i \nabla f(x_i)^T d_i \\ &= \frac{1}{2} \lambda_i^2 [d_i^T H_i^\epsilon d_i - \frac{1}{2} d_i^T (H_i^\epsilon + H_i^{-\epsilon}) d_i] - \frac{1}{12} \lambda_i^2 d_i^T (H_i^\epsilon + H_i^{-\epsilon}) d_i \end{aligned} \quad (4.12)$$

Since $x_i \rightarrow x^*$, then by Lemma 4.2, $d_i \rightarrow 0$. Thus H_i^1 , H_i^ϵ , and $H_i^{-\epsilon}$ converge to $H(x^*)$ and the first term in (4.12) will be less than $\frac{\gamma}{24} \lambda_i^2 \|d_i\|^2$ for i large enough. As in the proof of Theorem 4.1, $d_i^t (H_i^\epsilon + H_i^{-\epsilon}) d_i \geq \frac{\gamma}{2} \|d_i\|^2$ for large i . Substituting in (4.12), the desired result holds.

Now suppose that $\lambda_i > 1$ so that $\alpha_i = 1$. Then

$$f(x_i + \alpha_i d_i) - f(x_i) - \frac{1}{3} \alpha_i \nabla f(x_i)^t d_i = \frac{1}{2} d_i^t H_i^1 d_i + \frac{2}{3} \nabla f(x_i)^t d_i \quad (4.13)$$

Since $\lambda_i > 1$, then from (4.11) we must have

$$\nabla f(x_i)^t d_i < -\frac{1}{2} d_i^t (H_i^\epsilon + H_i^{-\epsilon}) d_i$$

Substituting in (4.13) we get:

$$\begin{aligned} f(x_i + \alpha_i d_i) - f(x_i) - \frac{1}{3} \alpha_i \nabla f(x_i)^t d_i &< \frac{1}{2} [d_i^t H_i^1 d_i - \frac{1}{2} d_i^t (H_i^\epsilon + H_i^{-\epsilon}) d_i] \\ &\quad - \frac{1}{12} d_i^t (H_i^\epsilon + H_i^{-\epsilon}) d_i \end{aligned} \quad (4.14)$$

That the right hand side of (4.14) is ≤ 0 for large i follows exactly in the same manner in which we proved that (4.12) is ≤ 0 . This completes the proof.

Finally, we state certain conditions in Theorem 4.3 below which guarantee that $\lambda_i < 1$ so that $\alpha_i = \lambda_i$ for i large enough.

Theorem 4.3

Let $\{x_i\}$ be a sequence generated by the algorithm. Suppose that $x_i \rightarrow x^*$ and that x^* satisfies the second order optimality conditions for Problem P'. If $\gamma < \frac{Y}{4}$, then there is an integer m so that $\lambda_i < 1$ for all $i \geq m$, that is, $\alpha_i = \lambda_i$ for all $i \geq m$.

Proof

By Theorem 4.1 there is an integer m so that for $i \geq m$ we have:

$$\lambda_i = \frac{-\varepsilon^2 \nabla f(x_i)^t d_i}{f(x_i + \varepsilon d_i) + f(x_i - \varepsilon d_i) - 2f(x_i)} = \frac{-\nabla f(x_i)^t d_i}{\frac{1}{2} d_i^t (H_i^\varepsilon + H_i^{-\varepsilon}) d_i} \quad (4.15)$$

As in the proof of Theorem 4.1

$$\frac{1}{2} d_i^t (H_i^\varepsilon + H_i^{-\varepsilon}) d_i \geq \frac{1}{4} \|d_i\|^2 \quad \text{for } i \text{ large enough} \quad (4.16)$$

Since w_i solves Problem $D(x_i)$, then there exist scalars $u_{ij} \geq 0$ for $j \in I(x_i)$ such that

$$\nabla f(x_i) + z w_i + \sum_{j \in I(x_i)} u_{ij} a_j = 0 \quad (4.17)$$

$$u_{ij} a_j^t w_i = 0 \quad \text{for } j \in I(x_i) \quad (4.18)$$

From (4.17) and (4.18) it follows that $\nabla f(x_i)^t w_i = -z \|w_i\|^2$. But by Theorem 3.1 x^* is a c-KT point and hence the optimal solution w^* to Problem $D(x^*)$ is $w^* = 0$. Since $x_i \rightarrow x^*$, by continuity of the optimal solution to Problem $D(\cdot)$, and since $b_j - a_j^t x_i > c$ for each $j \in I^+(w_i)$, it follows from (2.3) that $\beta_i = 1$ for large i . Thus $d_i = w_i$ so that

$$\nabla f(x_i)^t d_i = -z \|d_i\|^2 \quad \text{for large } i \quad (4.19)$$

Substituting (4.19) and (4.16) in (4.15), it is clear that $\lambda_i < 1$ for i large enough, and the proof is complete.

REFERENCES

1. L. Armijo, "Minimization of Functions Having Continuous Partial Derivatives," Pacific Journal of Mathematics, Volume 16, pp. 1-3, 1966.
2. M. S. Bazaraa and J. J. Goode, "An Algorithm for Finding the Shortest Element of a Polyhedral Set with Application to Lagrangian Duality," Journal of Mathematical Analysis and Applications, Volume 65, pp. 278-288, 1978.
3. J. W. Daniel, "Stability of the Solution of Definite Quadratic Programs," Mathematical Programming, Volume 5, pp. 41-53, 1973.
4. M. Frank and P. Wolfe, "An Algorithm for Quadratic Programming," Naval Research Logistics Quarterly, Volume 3, pp. 95-110, 1956.
5. P. E. Gill and W. Murray, "Newton-type Methods for Unconstrained and Linearly Constrained Optimization," Mathematical Programming, Volume 7, pp. 311-350, 1974.
6. D. Goldfarb, "Extension of Davidson's Variable Metric Method to Maximization Under Linear Inequality and Equality Constraints," SIAM Journal Applied Mathematics, Volume 17, pp. 739-764, 1969.
7. S. P. Han and O. L. Mangasarian, "Exact Penalty Functions in Nonlinear Programming," Mathematical Programming, Volume 17, pp. 251-269, 1979.
8. G. P. McCormick, "The Variable-Reduction Method for Nonlinear Programming," Management Science, Volume 17, pp. 146-160, 1970.
9. G. P. McCormick, "Second Order Conditions for Constrained Minima," SIAM Journal Applied Mathematics, Volume 15, pp. 641-652, 1967.
10. H. Mukai, "Readily Implementable Conjugate Gradient Methods," Mathematical Programming, Volume 17, pp. 298-319, 1979.
11. H. Mukai, "A Scheme for Determining Step Sizes for Unconstrained Optimization Methods," IEEE Transactions on Automatic Control, Volume AC23, pp. 987-995, 1978.
12. B. A. Murtagh and M. A. Saunders, "Large Scale Linearly Constrained Optimization," Mathematical Programming, Volume 14, pp. 41-72, 1978.
13. E. Polak, Computational Methods in Optimization: A Unified Approach, Academic Press, New York, 1971.
14. J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming, Part I, Linear Constraints," SIAM Journal Applied Mathematics, Volume 9, pp. 514-553, 1961.
15. D. M. Topkis and A. F. Veinott, "On the Convergence of Some Feasible Direction Algorithms for Nonlinear Programming," SIAM Journal Control Volume 5, pp. 263-279, 1967.

16. P. Wolfe, "Algorithm for a Least-Distance Programming Problem," Mathematical Programming Study 1, pp. 190-205, 1974.
17. P. Wolfe, "Methods of Nonlinear Programming," in Nonlinear Programming, edited by J. Abadie, North Holland Publishing Company, Amsterdam, 1967.
18. W. I. Zangwill, "The Convex Simplex Method," Management Science, Volume 14, pp. 221-283, 1967.
19. G. Zontendijk, Methods of Feasible Directions, Elsevier, Amsterdam, 1960.

